JVM是一个class文件的实现平台

Class文件是一种文件类型，是java文件经过编译形成的可供jvm识别执行的文件类型，以十六进制字节流的形式存在

基本包括 class文件标志，java版本号，

常量池：这里的常量不是只程序编写中的那些常量，而是包括程序所用到的所有字符，包括方法名，类的路径名字，也即之后所有会用到的字符

包含字面量和符号引用（标志位加指向其他字面量的索引构成一个类或其他的组织）

类的索引，父类的索引，接口索引集合

字段表

方法表

属性表

虚拟机类加载过程

加载，读取二进制字节流，并按照虚拟机格式储存在方法区中，在内存中实例化java.lang.class类的对象，作为方法区中加载类和外部的接口

验证

准备：准备阶段会正式为类的变量分配内存并设置类的变量的初始值（不包括非静态变量，只为static修饰的变量）注意：若public static int k =3;在准备阶段只会给予0的初始值，在初始化阶段才会赋予3；若public static final int k =3;在准备阶段就会给予3的初始值，这是个永远不会改变的常量。

—————————————————————————————————————到这里为止，已经可以看出在程序编译成class文件是只会把代码的调用入口，类的各种信息保存到里面，仅仅是二进制的字节流

而虚拟机在初始化开始前也仅仅是为加载类到方法区，处理静态变量和常量调用init方法

解析；将常量池中的符号引用替换为直接引用的过程

初始化：真正开始执行java中的程序代码，即所定义的赋值比较等的操作

虚拟机字节码执行引擎

换句话说就是怎样执行各种方法

Stack frame是虚拟机执行代码的方法里面包括

局部变量表，操作数栈（表示栈的深度），动态连接，方法地址，附加信息